root@kali: ~ — nmap

$ man nmap --cheat-sheet

Guía exhaustiva de Nmap

Network Mapper — descubrimiento de red y auditoría de seguridad. Manda paquetes a hosts/puertos y analiza las respuestas para saber qué hay vivo, qué puertos están abiertos, qué servicios y versiones corren, y qué sistema operativo.

SINTAXIS GENERAL

`nmap [tipo(s) de escaneo] [opciones] {especificación de objetivos}`

PERMISOS

Muchas técnicas (SYN scan, OS detection, escaneos raw) requieren **root / Administrador**. Sin privilegios, Nmap cae a TCP connect scan (`-sT`) por defecto.

LEGAL / ÉTICO

Escanear redes ajenas sin autorización explícita puede ser **ilegal** y/o detectado como ataque. Usá esto solo sobre infraestructura propia o con permiso por escrito.

// ÍNDICE

00 Mapa mental: el flujo de un escaneo
01 Especificación de objetivos
02 Host discovery
03 Técnicas de escaneo de puertos
04 Especificación de puertos
05 Detección de servicios y versiones
06 Detección de sistema operativo
07 NSE — Scripting Engine
08 Timing y rendimiento
09 Evasión de firewall / IDS
10 Output y reportes
11 Opciones varias útiles
12 Estados de puertos
13 Recetas combinadas
14 Flujo de trabajo recomendado
15 Errores comunes / gotchas
++ Apéndice: combos memorizables

ROOT requiere privilegios PELIGROSO intrusivo / puede romper LENTO tarda mucho

00 Mapa mental: el flujo de un escaneo

Nmap ejecuta sus fases internas en este orden. Entender este flujo explica casi todas las opciones: cada flag activa, desactiva o ajusta una fase.

#	Fase	Qué hace · flag relevante
1	**Enumeración de objetivos**	Resolver qué IPs vas a tocar.
2	**Host discovery**	¿Qué hosts están vivos? (ping scan). Se salta con `-Pn`.
3	**Resolución DNS**	DNS inverso de los hosts vivos. Se salta con `-n`.
4	**Port scanning**	Estado de los puertos. Se salta con `-sn`.
5	**Detección**	Versión (`-sV`), SO (`-O`), traceroute (`--traceroute`).
6	**NSE scripts**	Scripts en Lua (`-sC`, `--script`).
7	**Output**	Formateo y reporte de resultados.

01 Especificación de objetivos

Forma	Ejemplo	Qué hace
IP única	`nmap 192.168.1.1`	Un host.
Hostname	`nmap scanme.nmap.org`	Resuelve por DNS.
Rango octeto	`nmap 192.168.1.1-50`	.1 a .50.
Rango múltiple	`nmap 192.168.1-3.1-254`	Combina rangos por octeto.
CIDR	`nmap 192.168.1.0/24`	Subred entera (256 dir).
Comodín	`nmap 192.168.1.*`	Equivale a /24.
Lista de hosts	`nmap host1 host2 host3`	Varios separados por espacio.
Desde archivo	`nmap -iL targets.txt`	Una IP/host/rango por línea.
N aleatorios	`nmap -iR 100`	100 IPs públicas al azar (internet).
Excluir	`--exclude 192.168.1.5`	Saca host del barrido.
Excluir archivo	`--excludefile no.txt`	Saca una lista.
IPv6	`nmap -6 2001:db8::1`	Escaneo sobre IPv6.

`-iL` es tu mejor amigo en engagements reales: armás el alcance en un archivo y lo versionás. `--exclude` evita tocar máquinas fuera de alcance (gateways, prod sensible).

02 Host discovery

Sirve para saber qué hosts están vivos **antes** de gastar tiempo escaneando puertos de máquinas muertas.

Flag	Nombre	Uso directo
`-sn`	Ping scan	Solo descubre hosts vivos, NO escanea puertos. Reemplaza al viejo `-sP`.
`-Pn`	No ping	Asume que TODOS los hosts están vivos; se salta el discovery. Útil cuando el host no responde a ping pero sí tiene puertos abiertos (firewalls que dropean ICMP).
`-PS<p>`	TCP SYN ping	Manda SYN a esos puertos (ej. `-PS22,80,443`). Default 80.
`-PA<p>`	TCP ACK ping	Manda ACK; útil para evadir firewalls stateless.
`-PU<p>`	UDP ping	Bueno contra hosts que filtran TCP.
`-PY<p>`	SCTP INIT ping	Descubrimiento vía SCTP.
`-PE`	ICMP echo	Ping ICMP clásico (echo request).
`-PP`	ICMP timestamp	Algunos firewalls bloquean echo pero no timestamp.
`-PM`	ICMP netmask	Address mask request.
`-PR`	ARP ping	En LAN, lo más rápido y fiable; default automático en red local.
`-PO<pr>`	IP protocol ping	Manda paquetes de protocolos IP arbitrarios.
`-n`	Sin DNS	No resuelve nombres; **mucho más rápido**.
`-R`	Forzar DNS	Resuelve incluso hosts que parecen caídos.
`--dns-servers`	DNS custom	Usa servidores DNS específicos.
`--system-dns`	DNS del sistema	Usa el resolver del SO en vez del propio.
`--traceroute`	Traceroute	Traza la ruta a cada host.

```
$ // Barrido de inventario en LAN — lista qué está vivo, sin tocar puertos
$ nmap -sn 192.168.1.0/24
```

> En una LAN, Nmap usa ARP (`-PR`) por defecto porque ICMP/TCP pueden estar filtrados pero ARP no se puede ignorar dentro del mismo segmento. En internet/WAN combina ICMP echo + TCP SYN a 443 + TCP ACK a 80 + ICMP timestamp.

03 Técnicas de escaneo de puertos

El corazón de Nmap. Elegís UNA técnica TCP principal (algunas se pueden combinar).

Flag	Técnica	Cómo funciona · cuándo usar	Priv.
`-sS`	**TCP SYN** half-open / stealth	Manda SYN; si recibe SYN/ACK → abierto, manda RST sin completar handshake. **Default y recomendado.** Rápido, fiable, menos logueado.	ROOT
`-sT`	**TCP connect**	Completa el 3-way handshake vía `connect()`. Cuando NO tenés root. Más ruidoso (queda en logs).	NO
`-sU`	**UDP scan**	Manda datagramas UDP. Sin respuesta = abierto|filtrado. Para servicios UDP (DNS 53, SNMP 161, DHCP). LENTO	ROOT

Flag	Técnica	Cómo funciona · cuándo usar	Priv.
`-sA`	**TCP ACK**	Mapea reglas de firewall (filtrado / no filtrado, no abierto/cerrado). Auditar firewalls.	ROOT
`-sW`	**TCP Window**	Como ACK pero analiza el window size para inferir abierto/cerrado en algunos sistemas.	ROOT
`-sN`	**TCP Null**	Paquete sin flags. Evasión; explota RFC 793. No sirve contra Windows.	ROOT
`-sF`	**TCP FIN**	Solo flag FIN. Evasión sigilosa.	ROOT
`-sX`	**TCP Xmas**	FIN+PSH+URG ("árbol de navidad"). Evasión sigilosa.	ROOT
`--scanflags`	Flags custom	Definís vos los flags (ej. `--scanflags SYNFIN`). Evasión avanzada.	ROOT
`-sM`	**TCP Maimon**	FIN/ACK; explota bug en algunos BSD. Nicho.	ROOT
`-sY`	**SCTP INIT**	Equivalente a SYN para SCTP. Telecom / redes SCTP.	ROOT
`-sZ`	**SCTP COOKIE**	Más sigiloso en SCTP. Nicho.	ROOT
`-sO`	**IP protocol**	Determina qué protocolos IP (TCP, ICMP, IGMP…) soporta el host.	ROOT
`-sI`	**Idle scan**	100% indirecto usando un "host zombie" y predicción de IP ID. **No revela tu IP.** Avanzado.	ROOT
`-b`	**FTP bounce**	Abusa de servidores FTP viejos como proxy. Casi obsoleto (histórico).	NO

NULL / FIN / XMAS — POR QUÉ FUNCIONAN

Según RFC 793, un puerto **cerrado** responde RST a un paquete sin SYN/RST/ACK; uno **abierto** lo ignora. Por eso "sin respuesta = abierto|filtrado". Windows no sigue ese RFC → estas técnicas dan todo como abierto|filtrado en Windows (inútiles ahí).

SYN VS CONNECT — LA DIFERENCIA CLAVE

`-sS` no termina el handshake → la app del puerto muchas veces ni registra la conexión. `-sT` sí lo termina → aparece en los logs del servicio. Solo usás `-sT` obligado (sin root).

04 Especificación de puertos

Flag	Ejemplo	Qué hace
`-p <lista>`	`-p 22,80,443`	Puertos específicos.
Rango	`-p 1-1000`	Del 1 al 1000.
Todos	`-p-`	**Los 65535 puertos** (equiv. `-p 1-65535`).
Por protocolo	`-p T:80,U:53`	TCP 80 y UDP 53 (con `-sU -sS`).
Por nombre	`-p http,ssh`	Por nombre de servicio (de nmap-services).
Wildcard	`-p "http*"`	Todos los que matcheen.
Top N	`--top-ports 100`	Los N puertos más comunes.
Rápido	`-F`	"Fast": top 100 (en vez de top 1000 default).
Por frecuencia	`--port-ratio 0.05`	Puertos con frecuencia > 0.05.
Orden secuencial	`-r`	No randomiza el orden (default es aleatorio).

> **Default sin `-p`:** Nmap escanea los **1000 puertos más comunes** — no del 1 al 1000, sino los 1000 estadísticamente más frecuentes según `nmap-services`.

```
$ // Escaneo completo de un host — encuentra servicios en puertos altos
$ nmap -p- -sV 192.168.1.10
```

`-p-` es lento pero encuentra servicios escondidos en puertos altos (SSH en 2222, paneles en 8443). En recon serio se hace `-p-` al menos una vez por host.

05 Detección de servicios y versiones

Flag	Qué hace
`-sV`	Detecta **versión** del servicio (sondea banners, manda probes).
`--version-intensity <0-9>`	Intensidad: 0 = liviano/rápido, 9 = todos los probes. Default 7.
`--version-light`	= intensidad 2. Rápido, menos preciso.
`--version-all`	= intensidad 9. Exhaustivo, lento.
`--version-trace`	Debug detallado del proceso de detección.

```
$ // El combo clásico de recon
$ nmap -sV -p- 192.168.1.10
```

`-sV` es lo que convierte "puerto 443 abierto" en "nginx 1.18.0 sobre Ubuntu". Esa info de versión es la que cruzás con bases de CVE para encontrar vulnerabilidades. Sin `-sV` solo sabés que algo escucha, no qué.

06 Detección de sistema operativo

Flag	Qué hace
`-O`	Activa **OS fingerprinting** (analiza respuestas TCP/IP para inferir SO).
`-O --osscan-limit`	Solo detecta SO si hay ≥1 puerto abierto y 1 cerrado (más fiable, menos ruido).
`--osscan-guess` / `--fuzzy`	Adivina más agresivo cuando no hay match exacto.
`--max-os-tries <n>`	Limita los reintentos de detección de SO.
`-A`	**Agresivo:** equivale a `-O -sV -sC --traceroute` juntos.

```
$ // "Todo en uno": OS + versión + scripts default + traceroute
$ nmap -A 192.168.1.10
```

> La detección de SO necesita idealmente **un puerto abierto y uno cerrado** para comparar comportamientos. Si todo está filtrado, el resultado es poco fiable. Requiere root (manda paquetes raw).

07 NSE — Nmap Scripting Engine

El motor de scripts en Lua. Convierte a Nmap de scanner a framework de auditoría: detección de vulns, brute force, enumeración, etc.

> Activación

Flag	Qué hace
`-sC`	Corre los scripts de la categoría **default**.
`--script <nombre>`	Un script específico (ej. `--script http-title`).
`--script <categoría>`	Toda una categoría (ej. `--script vuln`).
`--script "<wildcard>"`	Comodín (ej. `--script "http-*"`).
`--script <a,b,c>`	Lista de scripts.
`--script-args`	Argumentos (ej. `user=admin,pass=1234`).
`--script-args-file`	Argumentos desde archivo.
`--script-help <n>`	Ayuda / descripción del script.
`--script-trace`	Debug de la comunicación del script.
`--script-updatedb`	Reconstruye la base de datos de scripts.

> Categorías de scripts

Categoría	Para qué
`auth`	Autenticación, bypass de credenciales.
`broadcast`	Descubrimiento por broadcast en LAN.
`brute` PELIGROSO	Fuerza bruta de credenciales (¡ruidoso!).
`default` SAFE	Set seguro y útil que corre con `-sC`.
`discovery`	Enumeración (servicios, directorios, hosts).
`dos` PELIGROSO	Denegación de servicio. Puede tirar el servicio.
`exploit` PELIGROSO	Explotación activa de vulns.
`external`	Usan recursos de terceros (envían datos afuera).
`fuzzer` PELIGROSO	Fuzzing de protocolos.
`intrusive` PELIGROSO	Agresivos, pueden crashear / alertar.
`malware`	Detectan backdoors / malware en hosts.

Categoría	Para qué
`safe` SAFE	No intrusivos, no deberían romper nada.
`version`	Extienden la detección de versión (`-sV`).
`vuln` CUIDADO	Chequean vulnerabilidades conocidas (CVEs).

> Scripts útiles concretos

```
# Banner grabbing
$ nmap --script banner 192.168.1.10

# Enumerar HTTP
$ nmap -p80,443 --script http-title,http-headers,http-enum 192.168.1.10

# Detección de vulnerabilidades conocidas
$ nmap --script vuln 192.168.1.10

# SMB: enumerar shares, usuarios, SO
$ nmap -p445 --script smb-os-discovery,smb-enum-shares,smb-enum-users 192.168.1.10

# SMB: chequeo EternalBlue (MS17-010)
$ nmap -p445 --script smb-vuln-ms17-010 192.168.1.10

# SSL/TLS: ciphers y vulnerabilidades
$ nmap -p443 --script ssl-enum-ciphers,ssl-heartbleed 192.168.1.10

# DNS zone transfer
$ nmap --script dns-zone-transfer --script-args dns-zone-transfer.domain=ejemplo.com -p53 ns.ejemplo.com

# Brute force SSH (ruidoso, solo con permiso)
$ nmap -p22 --script ssh-brute --script-args userdb=users.txt,passdb=pass.txt 192.168.1.10
```

CUIDADO CON LA CATEGORÍA

`-sC` (default) es seguro para correr casi siempre. `--script vuln` y `intrusive` / `exploit` / `dos` NO son seguros: pueden alertar, brutear cuentas o tirar servicios. Sabé qué categoría estás invocando.

Los scripts viven (típicamente) en `/usr/share/nmap/scripts/`. Podés listarlos ahí.

08 Timing y rendimiento

> Plantillas de timing (lo más usado)

Flag	Nombre	Velocidad	Cuándo
`-T0`	Paranoid	Lentísimo	Evasión IDS extrema (1 paquete cada varios minutos).
`-T1`	Sneaky	Muy lento	Evasión IDS.
`-T2`	Polite	Lento	Reduce carga / ancho de banda; no satura.
`-T3`	Normal	Default	Balance estándar.
`-T4`	Aggressive	Rápido	**Recomendado en redes rápidas / confiables.** Lo más usado.
`-T5`	Insane	Muy rápido	Sacrifica precisión por velocidad; puede perder puertos.

> Controles finos (override de las plantillas)

Flag	Qué controla
`--min/--max-hostgroup`	Cuántos hosts escanea en paralelo por grupo.
`--min/--max-parallelism`	N° de probes simultáneos.
`--*-rtt-timeout`	Tiempos de espera de respuesta (ej. `--max-rtt-timeout 500ms`).
`--max-retries <n>`	Reintentos por probe (default 10; bajarlo acelera).
`--host-timeout <t>`	Abandona un host si tarda más de X (ej. `30m`).
`--scan-delay <t>`	Espera fija entre probes (evasión / rate limit).
`--max-scan-delay <t>`	Tope del delay adaptativo.
`--min/--max-rate <n>`	Paquetes por segundo (ej. `--min-rate 1000`).

```
$ // Escaneo rápido y eficiente
$ nmap -T4 --min-rate 1000 -p- 192.168.1.10
```

> `-T4` ajusta varios parámetros a la vez (timeouts más cortos, más paralelismo). En LANs estables va perfecto. En redes con latencia o pérdida, `-T4` / `-T5` pueden reportar puertos abiertos como cerrados (respondieron tarde) → **falsos negativos**. Ante dudas, bajá a `-T3` y subí `--max-retries`.

09 Evasión de firewall / IDS / IPS

Flag	Técnica	Uso
`-f`	Fragmentar	Parte el header TCP en fragmentos de 8 bytes (`-ff` = 16). Confunde filtros viejos.

Flag	Técnica	Uso
`--mtu <n>`	MTU custom	Fragmenta a tamaño múltiplo de 8.
`-D <d1,ME,…>`	Decoys	Mezcla tu IP entre IPs falsas. `ME` marca tu posición. `-D RND:5` = 5 decoys aleatorios.
`-S <IP>`	Spoof source	Falsifica la IP origen (necesitás ver respuestas o no esperar resultados directos).
`-e <iface>`	Interfaz	Fuerza la interfaz de salida (útil con `-S`).
`--source-port` / `-g`	Puerto origen	Algunos firewalls confían en tráfico desde 53/80/443/20.
`--data-length <n>`	Padding	Bytes random para alterar la firma del paquete.
`--data <hex>`	Payload custom	Adjunta datos binarios a los paquetes.
`--data-string <s>`	Payload texto	Adjunta string custom.
`--ip-options`	IP options	Manipula opciones IP (source routing, etc.).
`--ttl <n>`	TTL custom	Setea el TTL.
`--spoof-mac <m>`	Spoof MAC	Falsifica la MAC (solo LAN). `0` = aleatoria; o por vendor (ej. `Cisco`).
`--badsum`	Checksum inválido	Checksums malos; respuestas suelen venir solo de IPS/IDS, no de hosts reales (los delata).
`--randomize-hosts`	Orden aleatorio	Dispersa el escaneo para no parecer secuencial.
`-sI <zombie>`	Idle scan	Anonimato total, no usás tu IP (ver sección 03).

```
$ // Escaneo evasivo
$ nmap -sS -T2 -f -D RND:5 --source-port 53 192.168.1.10
```

> La evasión real es un arte y los IDS modernos detectan la mayoría de estas técnicas. Los decoys (`-D`) son lo más práctico: el objetivo ve escaneos de varias IPs y no sabe cuál sos vos. El idle scan (`-sI`) es la única técnica donde tu IP nunca toca al objetivo, pero requiere un zombie con IP ID predecible (cada vez más raro).

10 Output y reportes

Flag	Formato	Uso
`-oN <f>`	Normal	Texto legible, igual que en pantalla.
`-oX <f>`	XML	Para parsear con scripts / herramientas.
`-oG <f>`	Greppable	Una línea por host; fácil de `grep` / `awk`. (Deprecado pero muy usado.)
`-oS <f>`	Script kiddie	"l33t". Broma, sin uso real.
`-oA <base>`	Todos	Genera `.nmap`, `.xml` y `.gnmap`. **Recomendado.**
`-v` / `-vv`	Verbose	Más detalle en tiempo real (escalable).
`-d` / `-dd`	Debug	Salida de depuración.

Flag	Formato	Uso
`--reason`	Razón	Muestra POR QUÉ Nmap marcó cada puerto así (ej. "syn-ack").
`--open`	Solo abiertos	Muestra únicamente puertos abiertos. **Muy útil.**
`--packet-trace`	Traza paquetes	Muestra cada paquete enviado / recibido.
`--append-output`	Append	Agrega a archivo existente en vez de sobrescribir.
`--stats-every <t>`	Stats	Imprime progreso cada X (ej. `10s`).
`--resume <f>`	Reanudar	Continúa un escaneo cortado.
`--noninteractive`	Sin teclas	Desactiva el control por teclado.

```
$ // Guardá SIEMPRE: genera recon_host10.nmap / .xml / .gnmap
$ nmap -A -oA recon_host10 192.168.1.10
```

TIP DURANTE EL ESCANEO

Apretá cualquier tecla para ver progreso; la barra espaciadora muestra estado; `v` / `d` suben verbosidad / debug en vivo.

11 Opciones varias útiles

Flag	Qué hace
`-6`	Habilita IPv6.
`-A`	Modo agresivo (`-O -sV -sC --traceroute`).
`--datadir <dir>`	Directorio de archivos de datos de Nmap.
`--send-eth` / `--send-ip`	Forzar envío en capa Ethernet o IP raw.
`--privileged` / `--unprivileged`	Asumir / forzar (no) privilegios.
`-V`	Versión de Nmap.
`-h`	Ayuda rápida.
`--iflist`	Lista interfaces y rutas (debug de red).
`--release-memory`	Libera memoria al terminar (debug).

12 Estados de puertos

Nmap clasifica cada puerto en uno de **6 estados**. Entenderlos es clave para interpretar resultados.

Estado	Significado
open	Una aplicación está aceptando conexiones. Es lo que buscás.

Estado	Significado
closed	El host responde, pero no hay app escuchando. Host vivo, puerto libre.
filtered	Nmap no puede determinar si está abierto: un firewall descarta los probes. Sin respuesta clara.
unfiltered	Accesible pero Nmap no sabe si abierto o cerrado (típico del ACK scan `-sA`).
open|filtered	No se pudo distinguir abierto de filtrado (típico de UDP, Null/FIN/Xmas).
closed|filtered	No se pudo distinguir cerrado de filtrado (típico de idle scan).

> `closed` ≠ `filtered`. **Closed** = host respondió RST (vivo, puerto sin servicio). **Filtered** = silencio o ICMP unreachable (algo bloquea). Esa distinción mapea las reglas del firewall. En UDP, "open|filtered" es lo normal: la ausencia de respuesta es ambigua — por eso UDP scan es lento y poco concluyente sin `-sV`.

13 Recetas combinadas (escenarios reales)

```
# 1. Inventario rápido de la red (qué hay vivo)
$ nmap -sn 192.168.1.0/24

# 2. Escaneo estándar de un host (top 1000 + versión + scripts default)
$ nmap -sV -sC 192.168.1.10

# 3. Recon completo de un host (todos los puertos + versión + SO + scripts)
$ nmap -p- -sV -sC -O -T4 -oA full_recon 192.168.1.10

# 4. Solo abiertos, rápido, con razón (triage inicial)
$ nmap -T4 --open --reason 192.168.1.0/24

# 5. UDP de los servicios UDP típicos
$ nmap -sU --top-ports 20 -sV 192.168.1.10

# 6. Escaneo TCP+UDP combinado
$ nmap -sS -sU -p T:1-1000,U:53,161,123 192.168.1.10

# 7. Búsqueda de vulnerabilidades (cuidado: intrusivo)
$ nmap -sV --script vuln -oA vulnscan 192.168.1.10

# 8. Web recon enfocado
$ nmap -p80,443,8080,8443 -sV --script "http-*" 192.168.1.10

# 9. Escaneo sigiloso / evasivo
$ nmap -sS -T2 -f -D RND:5 --source-port 53 -Pn 192.168.1.10

# 10. Barrido masivo rápido (muchos hosts)
```

```
$ nmap -sS -p- --min-rate 5000 -T4 -oG mass.gnmap 10.0.0.0/16

# 11. Host que no responde a ping pero está vivo
$ nmap -Pn -sV 192.168.1.10

# 12. Detección de SO fiable
$ nmap -O --osscan-limit --osscan-guess 192.168.1.10
```

14 Flujo de trabajo recomendado

Una secuencia típica en un pentest / auditoría:

#	Paso	Comando típico
1	**Discovery**	`nmap -sn <rango>` → lista de hosts vivos a un archivo.
2	**Triage de puertos**	`nmap -sS --top-ports 1000 -T4 -iL vivos.txt -oA triage`
3	**Escaneo profundo**	`nmap -p- -sV -sC -O -T4 -oA host_<ip> <ip>`
4	**UDP selectivo**	`nmap -sU --top-ports 50 -sV <ip>` → no olvidar UDP.
5	**Scripts dirigidos**	Según servicios encontrados (SMB, HTTP, SSL…).
6	**Vuln scan**	`--script vuln` sobre lo interesante.
7	**Documentar**	Todo con `-oA`, cruzar versiones con CVEs.

> PRINCIPIO CLAVE
>
> Escaneá de **rápido y amplio** a **lento y profundo**. Primero sabé qué hay; después invertí tiempo solo en lo que importa.

15 Errores comunes / gotchas

Gotcha	Por qué
Olvidar `sudo`	Sin root, `-sS` / `-O` / `-sU` no funcionan o caen a `-sT`. Resultados raros → fijate los privilegios.
No usar `-p-`	El default son solo los top 1000; servicios en puertos altos quedan invisibles.
Confiar en `-T5`	En redes lentas pierde puertos abiertos (falsos negativos). Usá `-T4` y subí retries.
Olvidar UDP	Muchos servicios críticos (DNS, SNMP, VPN) son UDP y el escaneo TCP no los ve.
`-Pn` indiscriminado	Se salta el discovery y escanea hosts muertos → lento. Solo cuando el host filtra ping.
`--script vuln` sin permiso	Es intrusivo y puede alertar / romper. No es "seguro".
No guardar output	Sin `-oA` perdés evidencia y tenés que re-escanear.

Gotcha	Por qué
Confundir `filtered` con `closed`	Son distintos. `filtered` = firewall; `closed` = vivo sin servicio.

++ Apéndice: combos memorizables

```
-sV -sC        → versión + scripts default (el combo de oro del recon)
-A             → todo agresivo (O + sV + sC + traceroute)
-p-            → los 65535 puertos
-T4            → rápido y confiable en LAN
--open         → solo lo que importa
--reason       → por qué Nmap dijo eso
-oA <base>     → guardá los 3 formatos siempre
-Pn            → no pinguees, asumí vivo
-sn            → solo descubrir, no escanear puertos
-iL <archivo>  → targets desde archivo
```

CIERRE

El 90% del trabajo real con Nmap se resuelve con un puñado de combos: `nmap -sV -sC -p- -T4 -oA salida <target>` para profundidad, y `nmap -sn <rango>` para descubrir. El resto de los flags son para casos específicos (evasión, UDP, scripts dirigidos, tuning). Dominá primero el flujo de 5 fases; los flags caen solos.

— EOF —